

MULTIMEDIA



UNIVERSITY

STUDENT ID NO

--	--	--	--	--	--	--	--	--	--

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2017/2018

TOP 2121 – OBJECT-ORIENTED PROGRAMMING (All sections / Groups)

11 OCTOBER 2017
9.00a.m. – 11.00a.m.
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 7 pages, including the cover page, with **FOUR** questions only.
2. Attempt all **FOUR** questions.
3. All questions carry equal marks and the distribution of the marks for each question is given.
3. Please write all your answers in the answer booklet provided.

Question 1

- (a) As a popular object-oriented programming language, Java has many great features. List **FOUR** non-OO features provided by Java.
[4 marks]
- (b) *Interface* and *Abstract* class are used to achieve the purpose of OO concepts in Java. Identify any **FOUR** differences between these two reference types.
[4 marks]
- (c) The output of the object-oriented analysis phase is intended to specify clearly what the system is supposed to do. List and describe the **TWO** outputs from analysis phase.
[2 marks]

Continued...

Question 2

- (a) Create an interface called Car that has methods to represent the standard operations (start, run and stop). Define the method signatures any way you desire. Describe how a class called MyCar might implement this interface.

[4 marks]

- (b) Write the output for the Java code as shown in Figure 1:

```
class Vehicle {  
    int petrol = 30;  
  
    public void move () {  
        System.out.println ("Vehicles are used to transport people  
                             or goods ");  
    }  
    public void petrol (int petrol) {  
        System.out.println ("Vehicles consume " + petrol + " liters  
                             petrol every day.");  
    }  
}  
class Car extends Vehicle {  
  
    public void move () {  
        super. move ();  
        System.out.println ("Car is a wheeled motor vehicle used  
                             for transportation ");  
    }  
    public void petrol (int petrol) {  
        System.out.println ("Vehicles consume " + super.petrol + " liters  
                             petrol every day.");  
    }  
}  
public class TestCar {  
    public static void main (String args []){  
        Vehicle v = new Vehicle();  
        Car c = new Car();  
        c.move();  
        v.move();  
        v.petrol(40);  
    }  
}
```

Figure 1. Sample Java Code.

[4 marks]

- (c) Explain the concept of inheritance in Java using the keyword *super* with an example.

[2 marks]

Continued...

Question 3

- (a) Translate the structure of Java code depicted in Figure 2 into a class diagram:

```
package HR;

class Employee {
    private String name;
    private double pay;
    private final String ID;

    public static final double START_PAY = ... ;
    public Employee(String name, double pay) { ... }
    public String getName() { ... }
    public int getID() { ... }
    public double getPay() { ... }
    public void changeName(String newName) { ... }
    public void changePay(double newPay) { ... }
}
```

Figure 2. A Sample Java Code.

[6 marks]

- (b) Given the scenario as follows, find out the necessary classes and draw a relationship diagram for the classes:
- Hospital has patients.
 - A Patient may have more than one illness.
 - Hospital distinguished Senior Patients from Normal patients, since Senior Patients enjoy subsidy from government.

[2 marks]

- (c) Draw a state diagram for the dialogue used for verification as shown in Figure 3.

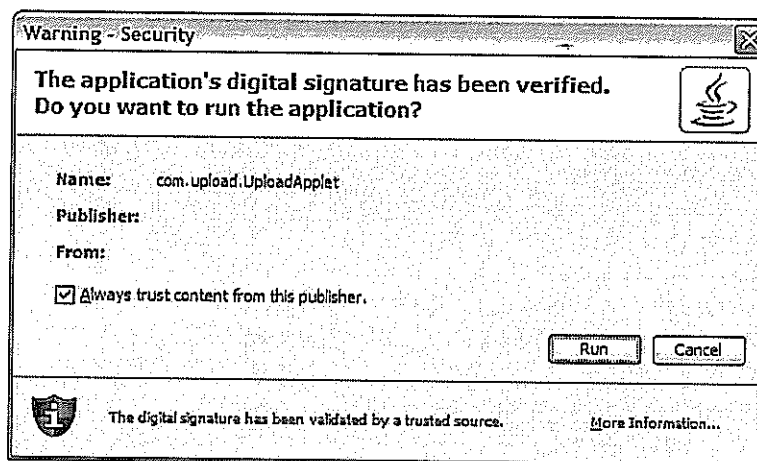


Figure 3. A Confirmation Dialogue

[2 marks]

Continued...

Question 4

- (a) Create a frame named as MyCalc as shown in the Figure 4 by completing the Java code given below using appropriate AWT components. The program used a panel for the Buttons and another panel for the Label and TextField. MyCalc does not provide any functionality to the components.

[3 marks]

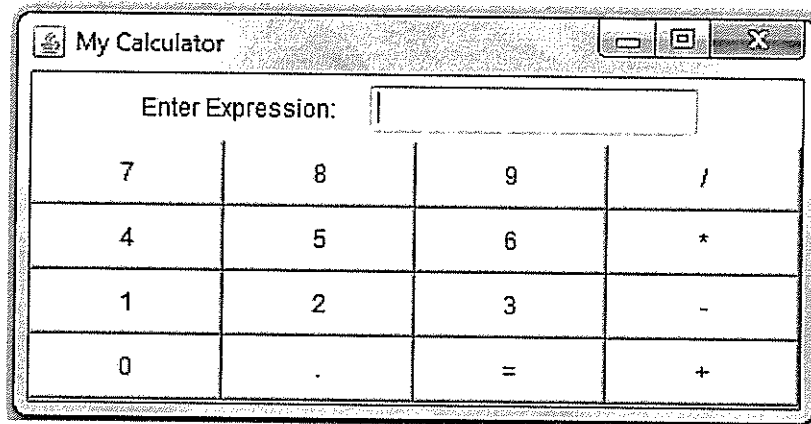


Figure 4. MyCalc

```
import java.awt.*;
import javax.swing.*;

public class MyCalc extends Frame{
    public static void main(String[] args){
        MyCalc m = new MyCalc();
        Panel p1=new Panel();
        p1.setLayout(new FlowLayout());
        p1.add(new Label("Enter Expression:"));
        p1.add(new TextField(20));
        Panel p2=new Panel();
        p2.setLayout(new GridLayout(4,4));
        p2.add(new Button ("7"));
        p2.add(new Button ("8"));
        p2.add(new Button ("9"));
        p2.add(new Button ("/"));
        p2.add(new Button ("4"));
        p2.add(new Button ("5"));
        p2.add(new Button ("6"));
        p2.add(new Button ("*"));
        p2.add(new Button ("1"));
        p2.add(new Button ("2"));
        p2.add(new Button ("3"));
        p2.add(new Button ("-"));
        p2.add(new Button ("0"));
        p2.add(new Button ((".")));
        p2.add(new Button ("="));
        p2.add(new Button ("+"));
    }
}
```

Continued...

```
m.add(p1, "North");
m.add(p2, "Center");
m.show();
}
}

MyCalc(){
    .... // complete the code
}
}
```

- (b) You are given a Java code as depicted in Figure 5. You are required to write the command to i) compile this code, and ii) execute the program to open a file called `sample.txt`.

```
import java.io.*;
public class ReadFile{
    public static void main(String args[]){
        int numRead = 0;
        FileReader reader = null;
        PrintWriter writer = null;
        char buffer[] = new char[512];
        reader = new FileReader (args [0]);
        writer = new PrintWriter(System.out);
        while ((numRead = reader.read(buffer)) != -1){
            writer.write(buffer, 0, numRead);
        }
        reader.close();
        writer.close();
    }
}
```

Figure 5. A Sample Java Code.

[2 marks]

Continued...

- (c) You are given a Java code as shown in Figure 6. Write down the missing class required to run the code.

```
class Drink extends Item{
    public Drink(String name, double price){
        super(name,price);
    }
}

class Inventory{
    public static void main(String args[]){
        Inventory i = new Inventory();
        Drink[] d = new Drink[2];
        d[0]=new Drink("Milo Ice", 2.0);
        d[1]=new Drink("Milo Hot", 1.5);

        for(int j=0;j<d.length;j++){
            System.out.println("Item "+j+ " is
                               "+d[j].getName()+". ");
            System.out.println("Its price is
                               RM"+d[j].getPrice()+". ");
        }
    }
}
```

Figure 6. Yet Another Sample Java Code.

[5 marks]

End of Paper